

WHAT IS CLAIMED IS:

1                   1.       A method for servicing interrupts generated by a plurality of co-  
2 processors included in a multiprocessor subsystem, the method comprising the acts of:  
3                   in response to a detected interrupt, determining whether the detected interrupt  
4 was generated by one of the plurality of co-processors of the multiprocessor subsystem; and  
5                   in the event that the detected interrupt was generated by one of the plurality of  
6 co-processors, scheduling execution of a deferred servicing procedure,  
7                   wherein during execution the deferred servicing procedure services a plurality  
8 of pending interrupts generated by one or more of the plurality of co-processors, including  
9 the detected interrupt.

1                   2.       The method of claim 1, wherein during execution the deferred  
2 servicing procedure services all pending interrupts from all of the plurality of processors.

1                   3.       The method of claim 1, wherein the plurality of pending interrupts  
2 serviced by the deferred servicing procedure includes a second interrupt generated by the one  
3 of the plurality of processors that generated the detected interrupt.

1                   4.       The method of claim 1, wherein the plurality of pending interrupts  
2 serviced by the deferred servicing procedure includes a second interrupt generated by one of  
3 the plurality of processors other than the one that generated the detected interrupt.

1                   5.       The method of claim 1, wherein the act of determining whether the  
2 detected interrupt was generated by one of the plurality of co-processors includes the acts of:  
3                   selecting one of the plurality of co-processors as a current co-processor; and  
4                   reading a value stored in an interrupt register of the current co-processor.

1                   6.       The method of claim 5, wherein in the event that the value stored in the  
2 interrupt register does not indicate an interrupt, a different one of the co-processors is  
3 selected and the act of reading is repeated.

1                   7.       The method of claim 5, wherein the act of reading includes:  
2                   updating a private register mapping to enable access to the interrupt register of  
3 the current co-processor,  
4                   wherein the private mapping is not used by the deferred servicing procedure.

1                    8.        The method of claim 1, further comprising the act of disabling further  
2 interrupts from the plurality of co-processors in the event that the detected interrupt was  
3 generated by one of the plurality of co-processors,  
4                    wherein during execution the deferred servicing procedure re-enables  
5 interrupts from the plurality of co-processors.

1                    9.        The method of claim 8, wherein the act of disabling further interrupts  
2 is performed at a critical priority level.

1                    10.     The method of claim 1, wherein the act of determining whether the  
2 detected interrupt was generated by one of the plurality of co-processors is performed at a  
3 critical priority level.

1                    11.     The method of claim 10, wherein the act of scheduling execution of the  
2 deferred servicing procedure is performed at the critical priority level.

1                    12.     The method of claim 11, wherein the act of scheduling execution of the  
2 deferred servicing procedure includes setting a second priority level for the deferred servicing  
3 procedure, wherein the second priority level is lower than the critical priority level.

1                    13.     The method of claim 1, wherein the multiprocessor subsystem is a  
2 graphics processing subsystem.

1                    14.     A computer system comprising:                    ✓  
2                    a multiprocessor subsystem including a plurality of co-processors for  
3 processing data, wherein each of the co-processors is configured to generate interrupts; and  
4                    a driver module configured to control operation of the multiprocessor  
5 subsystem, the driver module including:  
6                    a schedulable servicing module configured to detect and service all  
7 pending interrupts from all of the co-processors when activated; and  
8                    an interrupt detection module configured to schedule the servicing  
9 module for activation in the event of an interrupt from any one of the plurality  
10 of co-processors.

1                   15.     The system of claim 14, wherein the interrupt detection module is  
2 further configured to be activated by a central processing unit of the computer system in  
3 response to an interrupt signal.

1                   16.     The system of claim 14, wherein the interrupt detection module is  
2 further configured to disable further interrupts from all of the co-processors in the event of an  
3 interrupt from any one of the co-processors, and wherein the servicing module is further  
4 configured to re-enable further interrupts from all of the co-processors.

1                   17.     The system of claim 14, wherein the interrupt detection module is  
2 further configured to operate at a critical priority level and the servicing module is further  
3 configured to operate at a second priority level lower than the critical priority level.

1                   18.     The system of claim 14, wherein the multiprocessor subsystem is  
2 configured for graphics processing.

1                   19.     The system of claim 14, wherein each of the plurality of co-processors  
2 includes an interrupt register configured to indicate an interrupt, and wherein the interrupt  
3 detection module is further configured to determine whether one of the plurality of co-  
4 processors generated an interrupt by accessing the respective interrupt registers of the co-  
5 processors.

1                   20.     The method of claim 19, wherein the interrupt detection module is  
2 further configured to maintain a private mapping for accessing the respective interrupt  
3 registers, the private mapping being used exclusively by the interrupt detection module.

1                   21.     A computer program product comprising:  
2 a computer readable medium encoded with program code<sup>m</sup>, the program code  
3 including:

4                             program code for determining, in response to a detected interrupt,  
5 whether the detected interrupt was generated by one of the plurality of co-  
6 processors of the multiprocessor subsystem;

7                             program code for scheduling a deferred servicing procedure in the  
8 event that the detected interrupt was generated by one of the plurality of co-  
9 processors; and

10                    program code for performing the deferred servicing procedure,  
11                    wherein the program code for performing the deferred servicing procedure  
12                    includes program code for servicing a plurality of pending interrupts from one  
13                    or more of the plurality of processors, including the detected interrupt.